IN THE CLAIMS:

1.   (Currently Amended) A method in a data processing system for generating a generic compilation interface, from a first object-oriented software package, said method comprising the steps of:

identifying all public classes included in said first object-oriented software package;

for each of said public classes, identifying all public entities included in each of said public classes;

removing all references to software that is defined in a second object-oriented software package from said public entities included in each of said public classes;

generating an equivalent public class for each of said identified public classes, said equivalent public class including equivalent public entities that include no references to said software defined in said second object-oriented software package;

compiling each of said equivalent public classes; and

generating a compilation interface for said first object-oriented software package including each of said compiled equivalent public classes.

2.   (Original) The method according to claim 1, wherein said step of identifying all public entities included in each of said public classes further comprises the step of identifying all entities included in each of said public classes that include a public modifier.

3.   (Original) The method according to claim 1, further comprising the steps of:

determining whether each of said entities includes a native attribute;

in response to a determination that each of said entities includes a native attribute, removing said native attribute from each of said entities; and

generating equivalent entities that include no native attributes.

4.   (Original) The method according to claim 1, wherein the step of identifying all public entities included in each of said public classes further comprises the step of identifying all public methods included in each of said public classes.

5.   (Original) The method according to claim 1, wherein the step of identifying all public entities included in each of said public classes further comprises the step of identifying all public parameters included in each of said public classes.

6.   (Original) The method according to claim 1, wherein the step of identifying all public entities included in each of said public classes further comprises the step of identifying all public fields included in each of said public classes.

7.   (Currently Amended) The method according to claim 1, wherein said step of identifying all public classes included in said first object-oriented software package further comprises the step of identifying all public classes included in a Java Archive file.

8.   (Currently Amended) The method according to claim 1, wherein the step of identifying all public classes included in said first object-oriented software package further comprises the step of identifying all public classes included in said first object-oriented software package utilizing a java.util.jar utility.

9.   (Original) The method according to claim 1, wherein the step of for each of said public classes, identifying all public entities included in each of said public classes further comprises the step of for each of said public classes, identifying all public entities included in each of said public classes utilizing Java Reflection.

10.   (Original) The method according to claim 1, wherein said step of generating an equivalent public class for each of said identified public classes further comprises the step of generating a separate .java file for each of said identified public classes.

11.   (Original) The method according to claim 10, wherein said step of compiling each of said equivalent public classes further comprises the step of compiling each said .java file.

12. (Currently Amended) The method according to claim 11, wherein said step of generating a compilation interface for said first object-oriented software package including each of said compiled equivalent public classes further comprises the steps of:

gencrating a compilation Java Archive file; and

storing each said compiled .java file in said compilation Java Archive file.

13. (Original) The method according to claim 1, further comprising the step of utilizing said compilation interface within an Integrated Development Environment.

14. (Currently Amended) A data processing system for generating a generic compilation interface from a first object-oriented software package, said system comprising:

means for identifying all public classes included in said first object-oriented software package;

means for each of said public classes, for identifying all public entities included in each of said public classes;

means for removing all references to software defined in a second object-oriented software package from said public entities included in each of said public classes;

means for generating an equivalent public class for each of said identified public classes, said equivalent public class including equivalent public entities that include no references to said software defined in said second object-oriented software package;

means for compiling each of said equivalent public classes; and

means for generating a compilation interface for said first object-oriented software package including each of said compiled equivalent public classes.

15. (Original) The system according to claim 14, wherein said means for identifying all public entities included in each of said public classes further comprises means for identifying all entities included in each of said public classes that include a public modifier.

16.   (Original) The system according to claim 14, further comprising:

means for determining whether each of said entities includes a native attribute;

in response to a determination that each of said entities includes a native attribute, means for removing said native attribute from each of said entities; and

means for generating equivalent entities that include no native attributes.

17.   (Original) The system according to claim 14, wherein said means for identifying all public entities included in each of said public classes further comprises means for identifying all public methods included in each of said public classes.

18.   (Original) The system according to claim 14, wherein said means for identifying all public entities included in each of said public classes further comprises means for identifying all public parameters included in each of said public classes.

19.   (Original) The system according to claim 14, wherein said means for identifying all public entities included in each of said public classes further comprises means for identifying all public fields included in each of said public classes.

20.   (Currently Amended) The system according to claim 14, wherein said means for identifying all public classes included in said first object-oriented software package further comprises means for identifying all public classes included in a Java Archive file.

21.   (Currently Amended) The system according to claim 14, wherein said means for identifying all public classes included in said first object-oriented software package further comprises means for identifying all public classes included in said first object-oriented software package utilizing a java.util.jar utility.

22.   (Original) The system according to claim 14, wherein said means for each of said public classes, for identifying all public entities included in each of said public classes further comprises means for each of said public classes, for identifying all public entities included in each of said public classes utilizing Java Reflection.

23.   (Original) The system according to claim 14, wherein said means for generating an equivalent public class for each of said identified public classes further comprises means for generating a separate .java file for each of said identified public classes.

24.   (Original) The system according to claim 23, wherein said means for compiling each of said equivalent public classes further comprises means for compiling each said .java file.

25.   (Currently Amended) The system according to claim 24, wherein said means for generating a compilation interface for said first object-oriented software package including each of said compiled equivalent public classes further comprises:

means for generating a compilation Java Archive file; and

means for storing each said compiled .java file in said compilation Java Archive file.

26.   (Currently Amended) The system according to claim [[1]] 14, further comprising means for utilizing said compilation interface within an Integrated Development Environment.

27.   (Currently Amended) A computer program product in a data processing system for generating a generic compilation interface from a first object-oriented software package, said computer program product comprising:

instruction means for identifying all public classes included in said first object-oriented software package;

instruction means for each of said public classes, for identifying all public entities included in each of said public classes;

instruction means for removing all references to software defined in a second software object-oriented software package from said public entities included in each of said public classes;

instruction means for generating an equivalent public class for each of said identified public classes, said equivalent public class including equivalent public entities that include no references to said software defined in said second object-oriented

software package;

      instruction means for compiling each of said equivalent public classes; and

      instruction means for generating a compilation interface for said first object-oriented software package including each of said compiled equivalent public classes.

28.   (Original) The product according to claim 27, wherein said instruction means for identifying all public entities included in each of said public classes further comprises instruction means for identifying all entities included in each of said public classes that include a public modifier.

29.   (Original) The product according to claim 27, further comprising:

      instruction means for determining whether each of said entities includes a native attribute;

      instruction means in response to a determination that each of said entities includes a native attribute, for removing said native attribute from each of said entities; and

      instruction means for generating equivalent entities that include no native attributes.

30.   (Original) The product according to claim 27, wherein said instruction means for identifying all public entities included in each of said public classes further comprises instruction means for identifying all public methods included in each of said public classes.

31.   (Original) The product according to claim 27, wherein said instruction means for identifying all public entities included in each of said public classes further comprises instruction means for identifying all public parameters included in each of said public classes.

32.   (Original) The product according to claim 27, wherein said instruction means for identifying all public entities included in each of said public classes further comprises instruction means for identifying all public fields included in each of said public classes.

33. (Currently Amended) The product according to claim 27, wherein said instruction means for identifying all public classes included in said first object-oriented software package further comprises instruction means for identifying all public classes included in a Java Archive file.

34. (Currently Amended) The product according to claim 27, wherein said instruction means for identifying all public classes included in said first object-oriented software package further comprises instruction means for identifying all public classes included in said first object-oriented software package utilizing a java.util.jar utility.

35. (Original) The product according to claim 27, wherein said instruction means for each of said public classes, for identifying all public entities included in each of said public classes further comprises instruction means for each of said public classes, for identifying all public entities included in each of said public classes utilizing Java Reflection.

36. (Original) The product according to claim 27, wherein said instruction means for generating an equivalent public class for each of said identified public classes further comprises instruction means for generating a separate .java file for each of said identified public classes.

37. (Original) The product according to claim 36, wherein said instruction means for compiling each of said equivalent public classes further comprises instruction means for compiling each said .java file.

38. (Currently Amended) The product according to claim 37, wherein said instruction means for generating a compilation interface for said first object-oriented software package including each of said compiled equivalent public classes further comprises:

     instruction means for generating a compilation Java Archive file; and

     instruction means for storing each said compiled .java file in said compilation Java Archive file.

39.   (Original) The product according to claim 27, further comprising instruction means for utilizing said compilation interface within an Integrated Development Environment.